Management by Network Search

Misbah Uddin, Prof. Rolf Stadler
KTH Royal Institute of Technology, Sweden

Dr. Alex Clemm
Cisco Systems, CA, USA

November 11, 2014

ANRP Award Talks Session

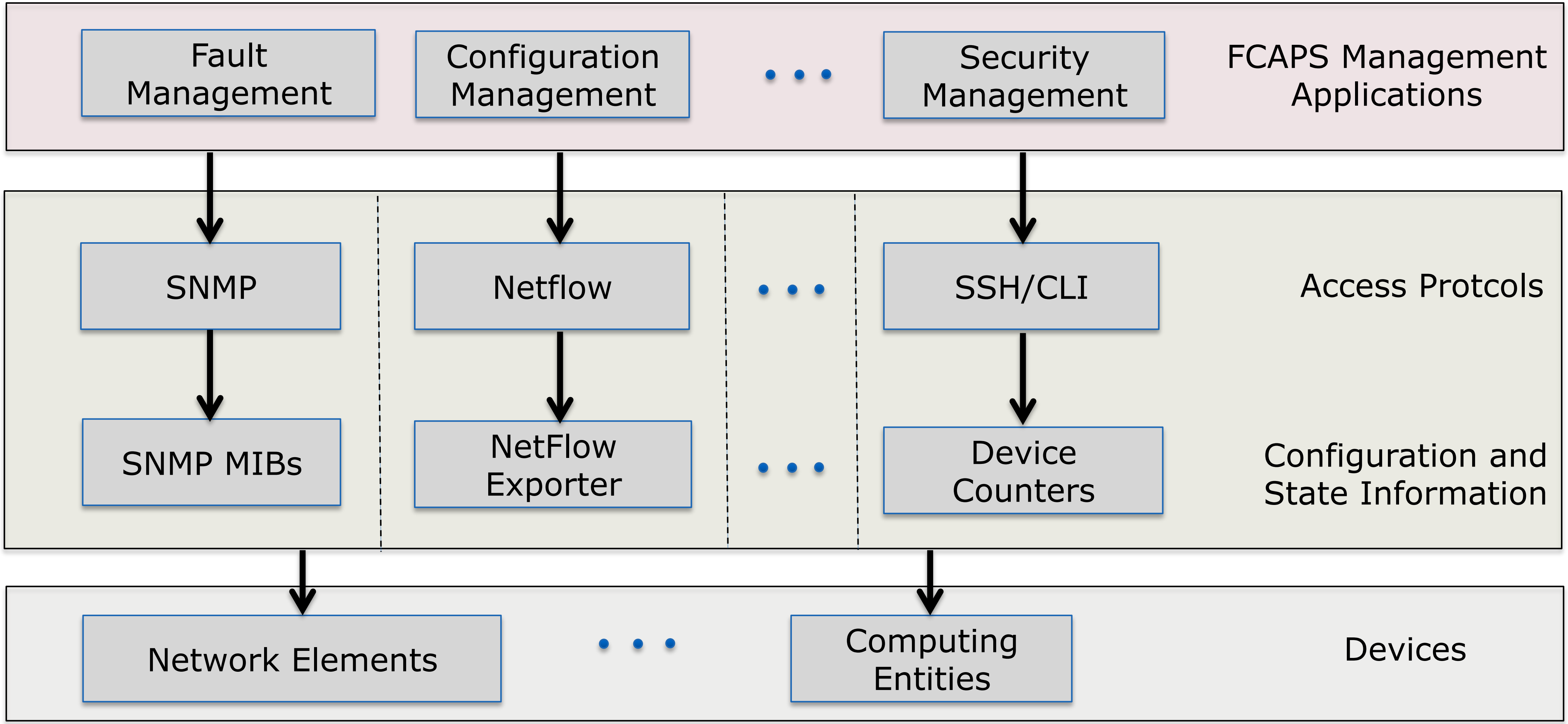
IETF 91 – Honolulu, Hawaii, November 9-14, 2014

ROYAL INSTITUTE
OF TECHNOLOGY

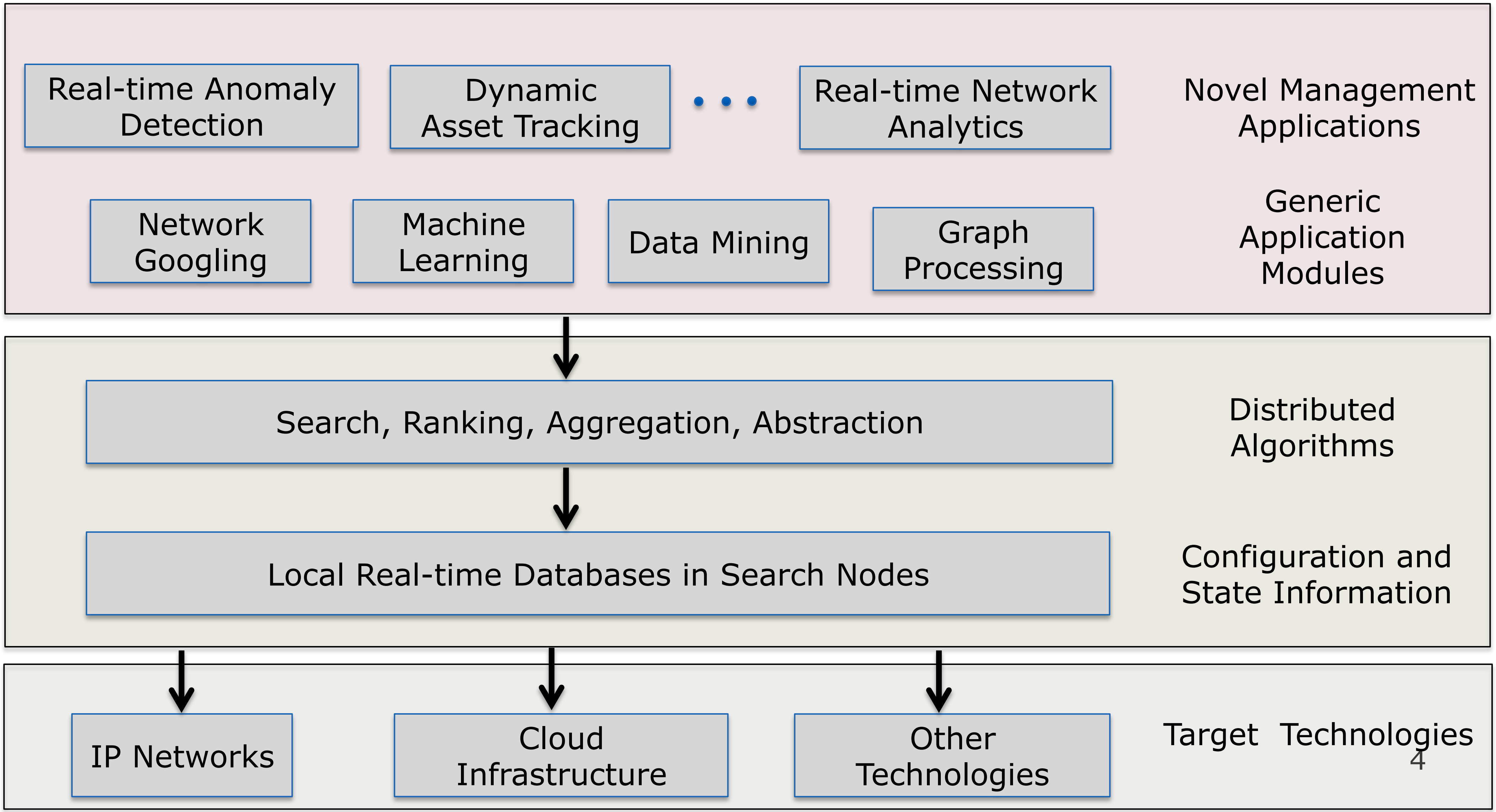
Outline

- Network Search: Motivation and context
- Model: Information, interface language, architecture
- Distributed processing of search queries
  Matching and ranking
- Search node design
- Performance of a prototype on a cloud testbed

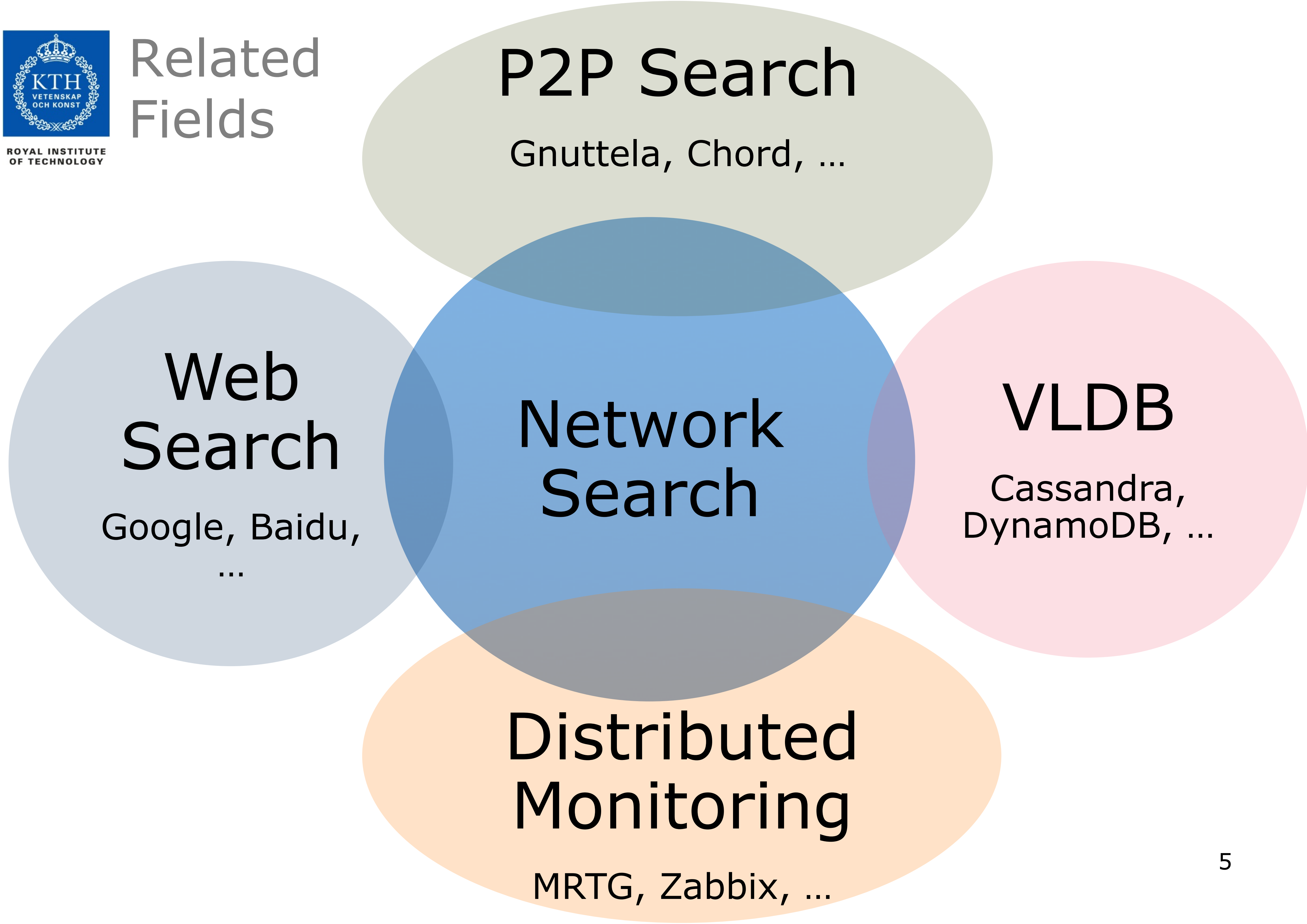
Traditional Management



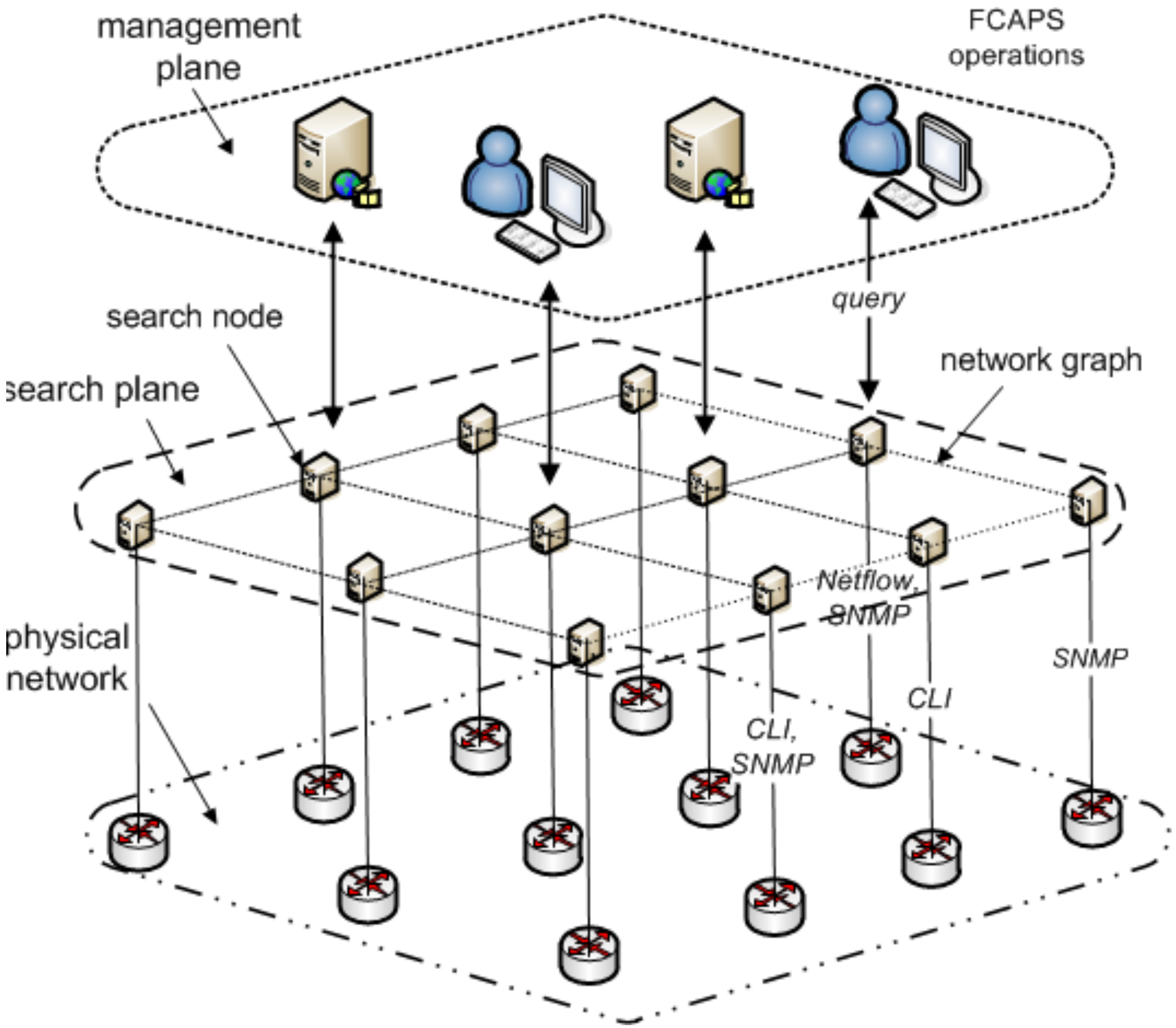
Management by Network Search



Related Fields



An Architecture for Network Search



Planes
- management plane
- search plane
- target technology

Search plane
- search node
- network graph

Search node
- real-time database
- data sensing
- various realizations

ROYAL INSTITUTE OF TECHNOLOGY

An Object Model for Network Search

object name	ns:instance-07
object type	VM
cpu core	2
memory	4 GB
ip address	10.10.5.33
search node	ns:cloud-01
uptime	35076 sec

object-name	ns:10.10.5.33:10.10.6.6
object-type	ip-flow
src port	24
dest port	93
bytes	2216
packets	23567
search node	ns:cloud-01
start time	14:12:35 May 20 2013

Existing information/data models for management are complex for search: GDMO, SMI, CIM, YANG, ontologies

Choose simple model for network search

- object: set of attribute-value pairs
- objects have a unique name (URN) and type
- relation: objects linked through joined attributes

A Query Language for Network Search

Token/ Search Term	T → A | V T → A op V	10.10.5.33 src-ip=10.10.5.33
Query	Q → T ∧ … ∧ T Q → T ∨ … ∨ T	10.10.5.33 10.10.6.6 cpu-load>0.8 *OR* memory-load>0.8
Link	λ (Q)	*link* (10.10.5.33 10.10.6.6)
Projection	$\Pi_{(A, \ldots, A)}$ (Q)	*project* (bandwidth, 10.10.5.33 10.10.6.6)
Aggregation	$\alpha_{(f, A)}$ (Q)	*max* (bandwidth, 10.10.5.33 10.10.6.6)

ROYAL INSTITUTE OF TECHNOLOGY

Matching and Ranking

Exact matching

- *M*: query, object → {0,1}
- used for databases

Approximate matching:

- *M*: query, object → score $\in$ [0,1]
- used for web search

Matching function *M* for network search

- based on extended Boolean retrieval model [Salton 83]
- supports exact and approximate matching
- considers occurrence of attributes in objects, and across object space

Matching and Ranking

Ranking function *R*

- maps query result into a list according to relevance
- *R*: $\{o_1, o_2, \ldots, o_n\} \rightarrow [o_{23} \geq o_4 \geq \ldots \geq o_{55}]$

Ranking function *R* for network search considers

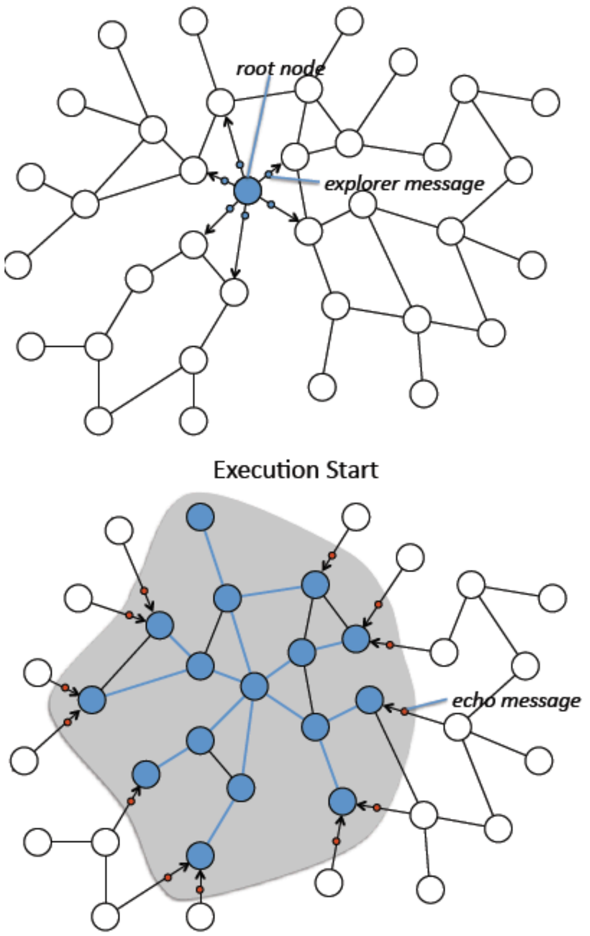
- matching score of an object
- connectivity of object
- freshness of information
- …

Relevance of an object to a search query depends on application.
(e.g., virtual asset tracking, root case analysis of fault, …)

→ Matching and ranking is parameterized in network search.

→ Design must support concurrent execution of queries with different matching and ranking semantics.

Distributed Processing of Search Queries



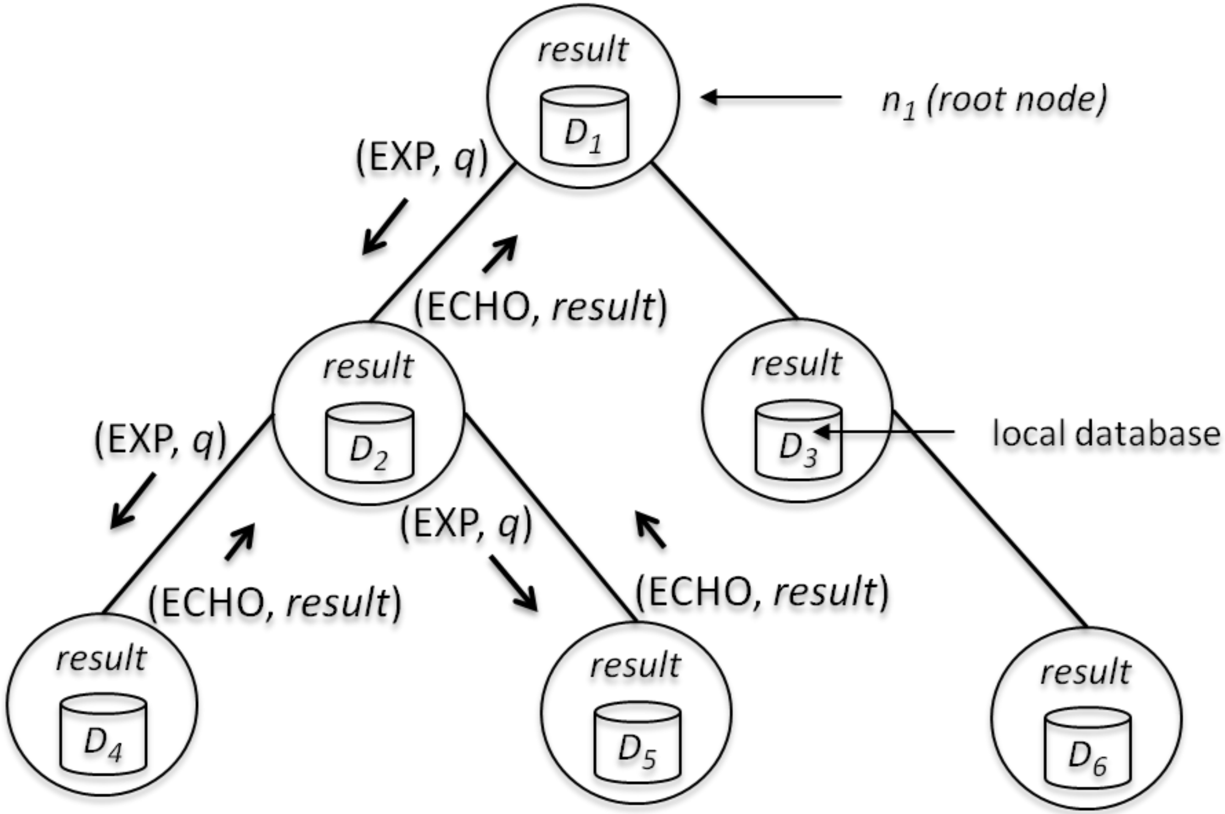
Execution Start
Expanding Wave
Contracting Wave
Echo Termination

Distributed algorithms on graph of search nodes

Design goal
- small latency
- low overhead
- scalability to >10^5 nodes

Approach
- Wave algorithm: Echo for query distribution
- Tree-based aggregation of partial results

Distributed Processing of Search Queries



On each search node:

- Matching query against local database
- Computing ranking score
- Aggregating partial results

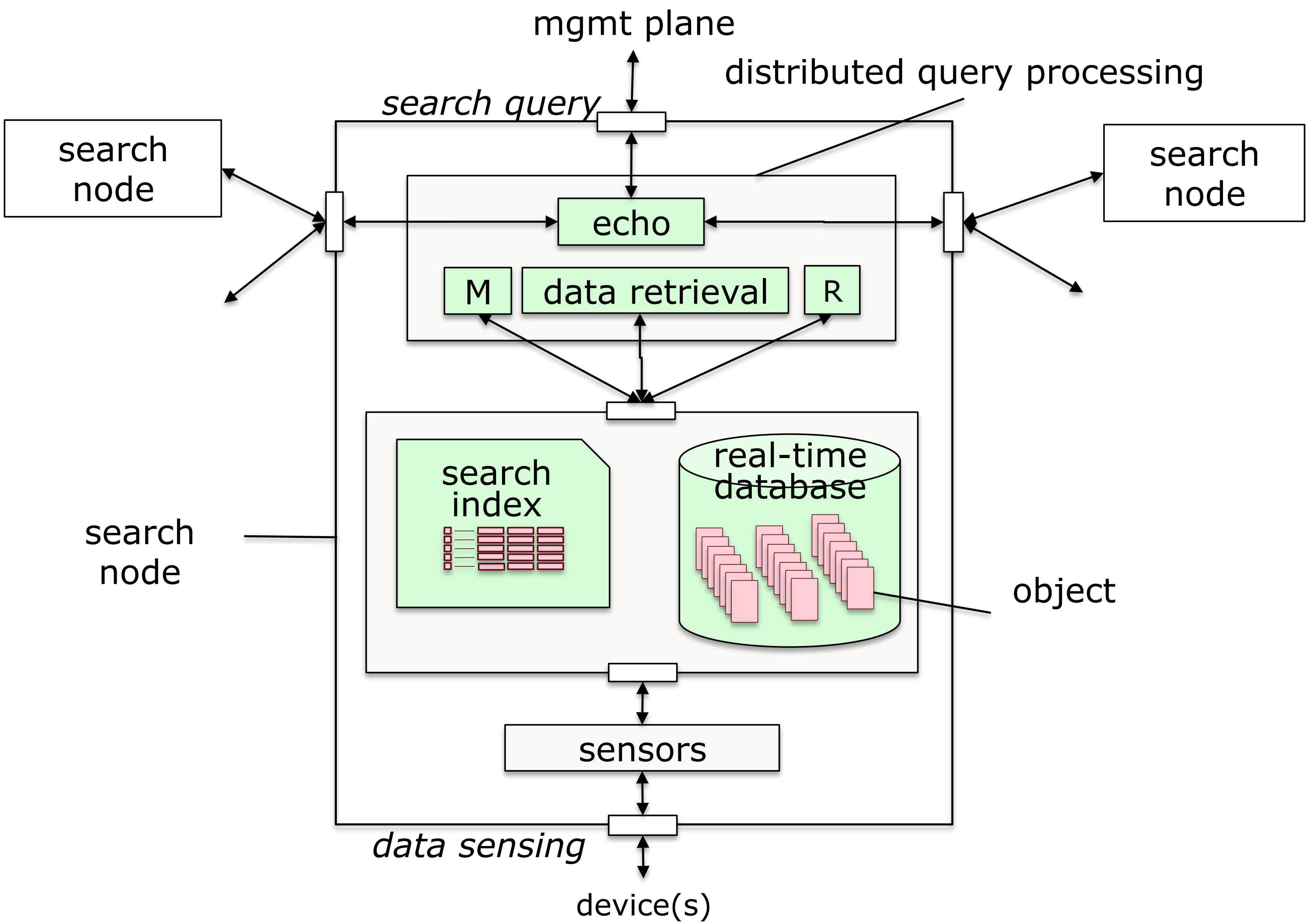
These functions are defined in an aggregator object.

Distributed Processing of Search Queries Aggregator

```
aggregator object processQuery( )
    var: qr : dictionary;
    procedure local( )
        qr := { };
        for each o ∈ M(q,D) do
            insert {name(o), o, R(q,o)} into qr;
        qr := top-k(qr);
    procedure aggregate(child-qr: dictionary)
        qr := top-k(merge(qr, child-qr));
```

State of aggregator:
qr contains tuples *(object name, object, ranking score)*

Design of a Search Node

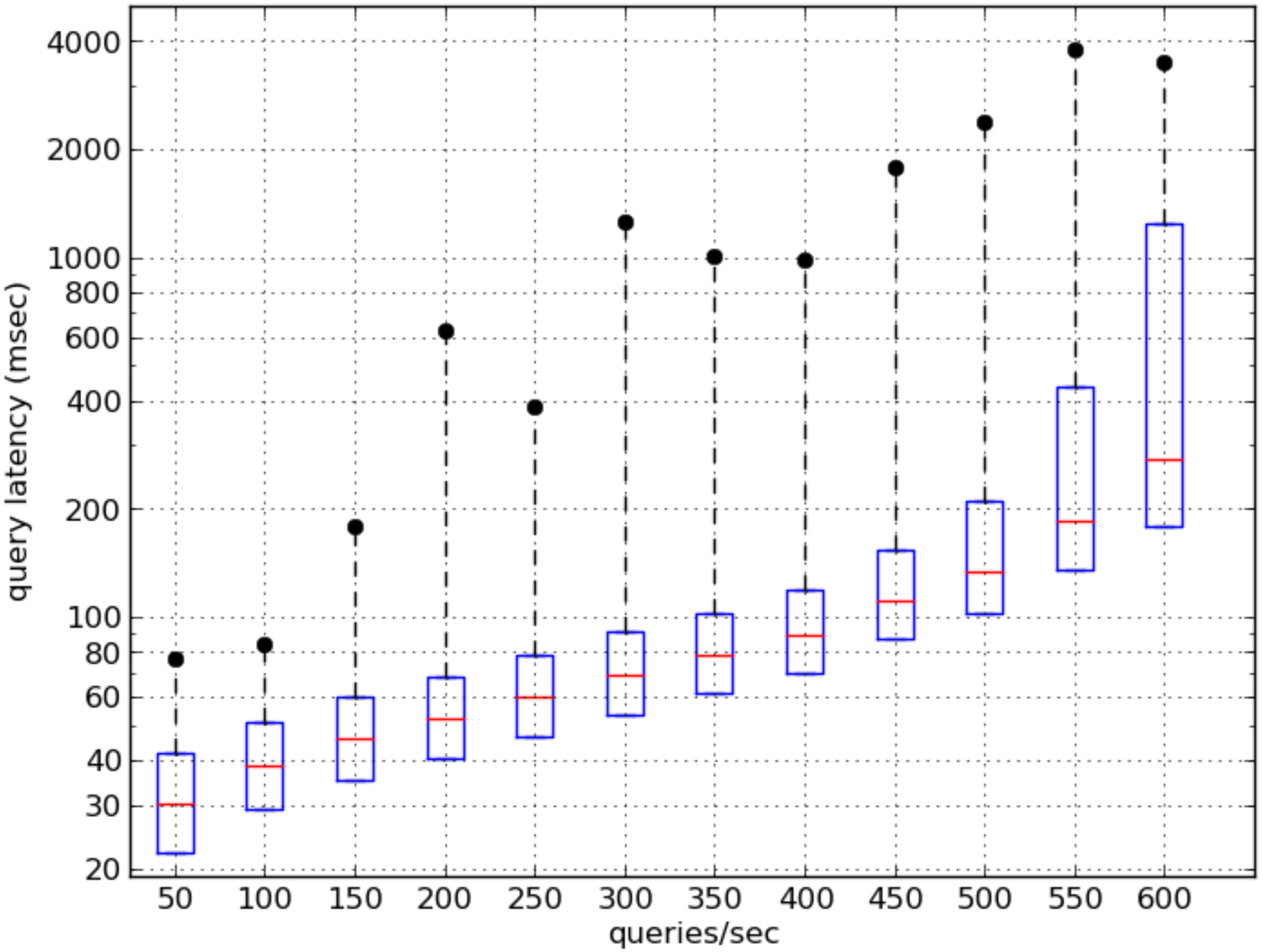


ROYAL INSTITUTE OF TECHNOLOGY

Cloud Testbed for Evaluation

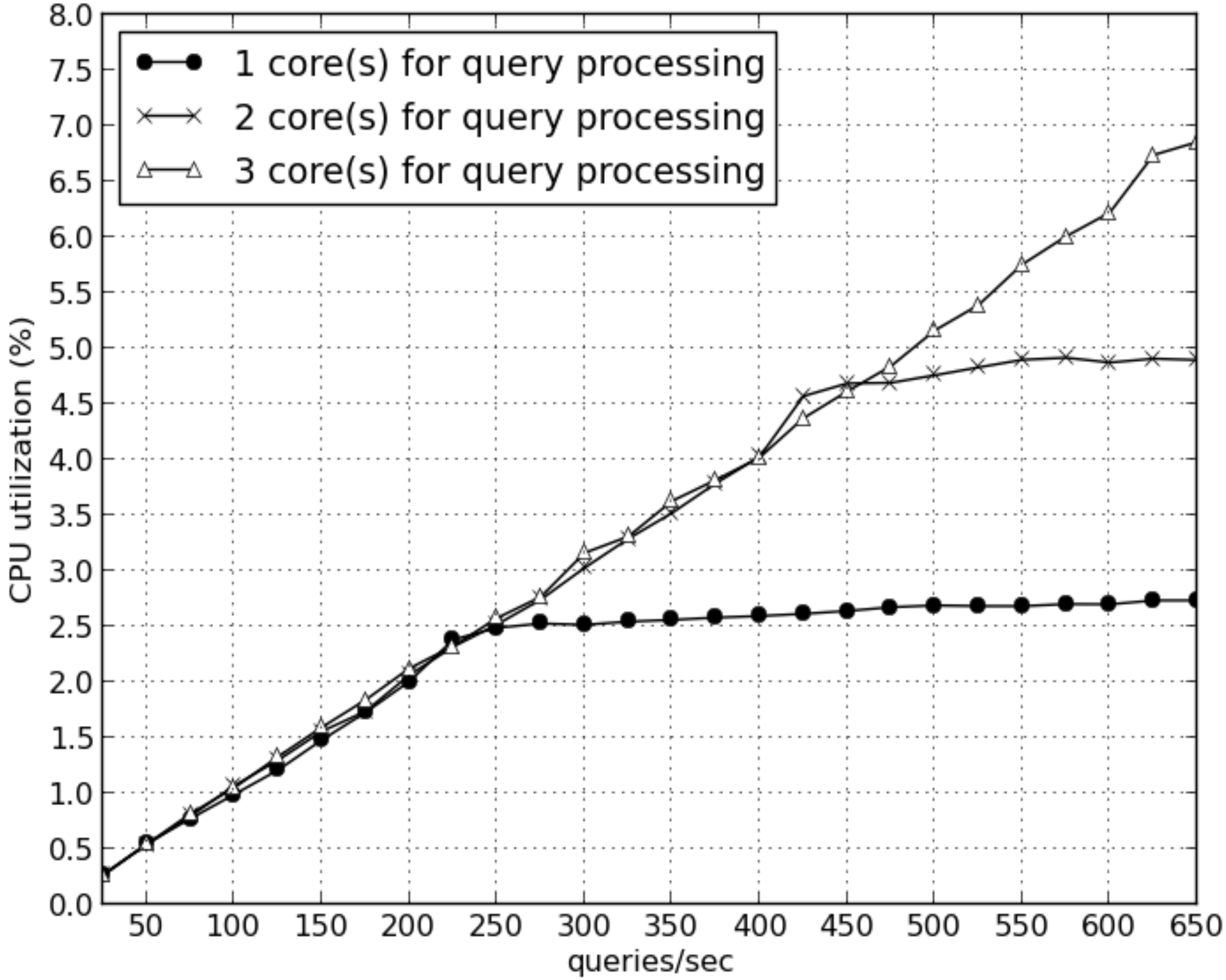
- Hardware
  - 9 Dell PowerEdge servers (24 cores/64GB)
  - Switched GB Ethernet
- Software
  - Ubuntu
  - OpenStack
  - Search Node on each server
- Query Load
  - global search queries with 2-5 tokens
  - 75% queries, 25% updates
  - Poisson arrivals
- Evaluation Metrics
  - Query latency
  - CPU utilization

Performance of the Prototype
Latency of global queries



25, 50, 75, 95th percentile value.
Nodes run 3 query processors, 90% CPU load on servers.

Performance of the Prototype
Computational overhead



ROYAL INSTITUTE OF TECHNOLOGY

Some Open Questions

- Spatial Search: *search for objects with spatial properties, e.g., within a radius of another object*
- Scalability: *reduce search space for given query*
- Persistent search: *search on streaming data*
- Historical Data: *search for past states and objects*
- Security/Privacy
- Multi-domain Search

ROYAL INSTITUTE OF TECHNOLOGY

Contribution

- Network Search is based on an information-centric view of network management.

  It supports real-time management.

  It is suited for large, dynamic networked systems.

- Presented a simple, but complete model:
  architecture, object model, query language
  distributed processing of search queries
  search node design

- Building a network search system for a network and cloud infrastructure is feasible.

ROYAL INSTITUTE OF TECHNOLOGY

Literature

- M. Uddin, R. Stadler, A. Clemm: Management by Network Search, NOMS 2012.
- M. Uddin, R. Stadler, A. Clemm: A Query Language for Network Search, IM 2013.
- M. Uddin, A. Skinner, R. Stadler, A. Clemm: Real-Time Search in Clouds, Demonstration, IM 2013.
- M. Uddin, R. Stadler, A. Clemm: Scalable Matching and Ranking for Network Search, CNSM 2013.
- M. Uddin, R. Stadler, M. Miyazawa, M. Hayashi: Graph Search for Cloud Network Management, IM 2014